



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/966,131	09/27/2001	Vijay Seetharaman	JP920000314US1	4515
26502	7590	07/30/2004	EXAMINER	
IBM CORPORATION			PESIN, BORIS M	
IPLAW IQ0A/40-3				
1701 NORTH STREET			ART UNIT	PAPER NUMBER
ENDICOTT, NY 13760			2174	
DATE MAILED: 07/30/2004				

6

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>
	09/966,131	SEETHARAMAN ET AL.
	<b>Examiner</b>	<b>Art Unit</b>
	Boris Pesin	2174

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) Responsive to communication(s) filed on \_\_\_\_\_.
- 2a) This action is FINAL.                            2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) Claim(s) 1-13 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 1-13 is/are rejected.
- 7) Claim(s) \_\_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  - a) All    b) Some \* c) None of:
    1. Certified copies of the priority documents have been received.
    2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
    3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)  | Paper No(s)/Mail Date. _____ .  |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ . | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
|   | 6) <input type="checkbox"/> Other: _____ .                                  |

## DETAILED ACTION

### ***Claim Rejections - 35 USC § 102***

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-13 are rejected under 35 U.S.C. 102(e) as being anticipated by Bowman-Amuah (US 6636424).

In regards to claim 1, Bowman-Amuah teaches a computer software system comprising: a view sub-system including presentation objects which provide a user interface (i.e. Figure 42, "User Interface Component"); a business logic sub-system including business object implementation objects which hold business data objects and implement business functions (i.e. "*Business Logic is the core of any application, providing the expression of business rules and procedures (e.g., the steps and rules that govern how a sales order is fulfilled)*). As such, the *Business Logic includes the control structure that specifies the flow for processing business events and user requests. There are many ways in which to organize Business Logic, including: rules-based, object-oriented, components, structured programming, etc. however each of these techniques include, although perhaps not by name, the concepts of: Interface, Application Logic, and Data Abstraction. FIG. 33 depicts the various components of the Business Logic portion of the Netcentric Architecture Framework.*" Column 119, Line 53); a handler sub-system including controller objects which control a sequence of actions in a use case (i.e. "*FIG. 158 illustrates a flowchart for a method 15800 for*

*controlling data. A data retrieval mechanism is provided in operation 15802 for retrieving data from a database. The data retrieval mechanism also writes data to the database. In operation 15804, the data retrieval mechanism is encapsulated in a data handler. A request from a domain object is received for a retrieval of a portion of the data in the database in operation 15806. The data retrieval mechanism is utilized in operation 15808 to retrieve the portion of the data from the database. The portion of the data is passed through the data handler to the domain object in operation 15810."*

Column 275, Line 56); and a view context sub-system including at least one context object which is arranged to capture input and output data (i.e. "The Client asks the Browsing Interface Proxy for the data associated with customer 1234. 6. The Browsing Interface Proxy forwards the request across the network to the Browsing Interface. 7. Same as 6. 8. The request is forwarded to the Customer Server. The Customer Server requests the customer data from the Database. 9. The Database returns the customer data for Customer 1234. 10. The Customer Server creates a structure and populates it with the customer data. 11. The Customer Structure is forwarded through the Browsing Interface, across the network and back to the Browsing Interface Proxy. 12. The Browsing Interface Proxy forwards the Customer Structure to the Client. The Client can now display the data in a UI for a user." Column 213, Line 34).

In regards to claim 2, Bowman-Amuah teaches a computer software system wherein: the view context sub-system is arranged to capture the input and output data which populates the presentation objects of the view sub-system (i.e. "The Client asks the Browsing Interface Proxy for the data associated with customer 1234. 6. The

*Browsing Interface Proxy forwards the request across the network to the Browsing Interface. 7. Same as 6. 8. The request is forwarded to the Customer Server. The Customer Server requests the customer data from the Database. 9. The Database returns the customer data for Customer 1234. 10. The Customer Server creates a structure and populates it with the customer data. 11. The Customer Structure is forwarded through the Browsing Interface, across the network and back to the Browsing Interface Proxy. 12. The Browsing Interface Proxy forwards the Customer Structure to the Client. The Client can now display the data in a UI for a user.” Column 213, Line 34).*

In regards to claim 3, Bowman-Amuah teaches a computer software system wherein: the view context sub-system also comprises data interfaces for the business logic sub-system (i.e. Figure 33, Elements 3300, 3302, and 3304).

In regards to claim 4, Bowman-Amuah teaches a computer software system wherein: the context objects included in the view context sub-system are updated when input is entered into the view sub-system by a user (i.e. “*an application may be designed to retrieve the tax rate for the State of Illinois. When the user enters “Illinois” on the screen, the application first validates the user’s entry by checking for its existence on the “State Tax Table”, and then retrieves the tax rate for Illinois. Note that codes tables provide an additional degree of flexibility. If the tax rates changes, the data simply needs to be updated; no application logic needs to be modified.*” Column 102, Line 33); and the context data objects are updated by the handler sub-system whenever business logic is executed on any of the context objects (i.e. “*FIG. 158*

*illustrates a flowchart for a method 15800 for controlling data. A data retrieval mechanism is provided in operation 15802 for retrieving data from a database. The data retrieval mechanism also writes data to the database. In operation 15804, the data retrieval mechanism is encapsulated in a data handler. A request from a domain object is received for a retrieval of a portion of the data in the database in operation 15806. The data retrieval mechanism is utilized in operation 15808 to retrieve the portion of the data from the database. The portion of the data is passed through the data handler to the domain object in operation 15810." Column 275, Line 56).*

In regards to claim 5, Bowman-Amuah teaches a computer software system wherein: the view sub-system refreshes the presentation objects with the input and output data from the view context sub-system (i.e. "A *retrieveData( )* message might also be required, if the object model pre-instantiates objects before retrieving them. Similarly, *refresh( )* could be used: the business object, if dirty, replaces any changes with data originally from the data store." Column 296, Line 13).

In regards to claim 6, Bowman-Amuah teaches a computer software system wherein: the view context sub-system is represented in a platform-independent format (i.e. "The Directory service can organize network nodes to reflect the topology and organization of the enterprise and its policies. The Directory service makes resources location and platform independent, since it allows users to locate resources via the directory and regardless of their physical location. The Directory service also maps between logical resource names" Column 62, Line 57).

In regards to claim 7, Bowman-Amuah teaches a computer program comprising: at least one view object comprising presentation objects which provide a user interface (i.e. Figure 42, "User Interface Component"); at least one business logic object comprising business data objects and arranged to implement business functions (i.e. *"Business Logic is the core of any application, providing the expression of business rules and procedures (e.g., the steps and rules that govern how a sales order is fulfilled). As such, the Business Logic includes the control structure that specifies the flow for processing business events and user requests. There are many ways in which to organize Business Logic, including: rules-based, object-oriented, components, structured programming, etc. however each of these techniques include, although perhaps not by name, the concepts of: Interface, Application Logic, and Data Abstraction.* FIG. 33 depicts the various components of the Business Logic portion of the Netcentric Architecture Framework." Column 119, Line 53); at least one handler object which controls actions of at least one of the view objects and actions of at least one of the business objects (i.e. "*FIG. 158 illustrates a flowchart for a method 15800 for controlling data. A data retrieval mechanism is provided in operation 15802 for retrieving data from a database. The data retrieval mechanism also writes data to the database. In operation 15804, the data retrieval mechanism is encapsulated in a data handler. A request from a domain object is received for a retrieval of a portion of the data in the database in operation 15806. The data retrieval mechanism is utilized in operation 15808 to retrieve the portion of the data from the database. The portion of the data is passed through the data handler to the domain object in operation 15810.*"

Column 275, Line 56); and at least one view context object comprising data objects which capture a state of at least one of the view objects (i.e. “*For instance, all of the constants used by a PhoneNumber object to capture the various types of PhoneNumber (i.e. home, business, fax, cell, pager, etc) can be accessed through a PhoneTypeConstants class.*” Column 203, Line 3).

In regards to claim 8, Bowman-Amuah teaches a computer program according wherein: each view context object is associated with a single view object; and the view context object is arranged to capture all data objects needed to populate the presentation objects of the associated view object at any one time (i.e. “*The Client asks the Browsing Interface Proxy for the data associated with customer 1234. 6. The Browsing Interface Proxy forwards the request across the network to the Browsing Interface. 7. Same as 6. 8. The request is forwarded to the Customer Server. The Customer Server requests the customer data from the Database. 9. The Database returns the customer data for Customer 1234. 10. The Customer Server creates a structure and populates it with the customer data. 11. The Customer Structure is forwarded through the Browsing Interface, across the network and back to the Browsing Interface Proxy. 12. The Browsing Interface Proxy forwards the Customer Structure to the Client. The Client can now display the data in a UI for a user.*” Column 213, Line 34).

In regards to claim 9, Bowman-Amuah teaches a computer program wherein: each view context object also comprises data interfaces for the business logic objects

accessed in a use case in which the associated view participates (i.e. Figure 33, Elements 3300, 3302, and 3304).

In regards to claim 10, Bowman-Amuah teaches a computer wherein: the data objects associated with a view context object are updated when input is entered into the associated view object by a user (i.e. “*an application may be designed to retrieve the tax rate for the State of Illinois. When the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the "State Tax Table", and then retrieves the tax rate for Illinois. Note that codes tables provide an additional degree of flexibility. If the tax rates changes, the data simply needs to be updated; no application logic needs to be modified.*” Column 102, Line 33); and data elements are updated by a handler object whenever business logic is executed on the data elements (i.e. “*FIG. 158 illustrates a flowchart for a method 15800 for controlling data. A data retrieval mechanism is provided in operation 15802 for retrieving data from a database. The data retrieval mechanism also writes data to the database. In operation 15804, the data retrieval mechanism is encapsulated in a data handler. A request from a domain object is received for a retrieval of a portion of the data in the database in operation 15806. The data retrieval mechanism is utilized in operation 15808 to retrieve the portion of the data from the database. The portion of the data is passed through the data handler to the domain object in operation 15810.*” Column 275, Line 56).

In regards to claim 11, Bowman-Amuah teaches a computer program wherein: the view object associated with a view context object is refreshed with the data objects

associated with a view context object (i.e. “*A retrieveData( ) message might also be required, if the object model pre-instantiates objects before retrieving them. Similarly, refresh( ) could be used: the business object, if dirty, replaces any changes with data originally from the data store.*” Column 296, Line 13).

In regards to claim 12, Bowman-Amuah teaches a computer program according wherein: the at least one view context object is represented in a separate platform-independent format (i.e. “*The Directory service can organize network nodes to reflect the topology and organization of the enterprise and its policies. The Directory service makes resources location and platform independent, since it allows users to locate resources via the directory and regardless of their physical location. The Directory service also maps between logical resource names*” Column 62, Line 57).

In regards to claim 12, Bowman-Amuah teaches a method of passing data in an object oriented application having at least one handler object, the method comprising the steps of: creating a view object with the handler object (i.e. “*Interface logic interprets and maps the actions of users into business logic processing activities. With the assistance of Presentation Services, Interface logic provides the linkage that allows users to control the flow of processing within the application.*” Column 119, Line 66); creating a view context object with the view object (i.e. Figure 42, “User Interface Component”); passing the view context object to the handler object (i.e. “*FIG. 158 illustrates a flowchart for a method 15800 for controlling data. A data retrieval mechanism is provided in operation 15802 for retrieving data from a database. The data retrieval mechanism also writes data to the database. In operation 15804, the data*”

*retrieval mechanism is encapsulated in a data handler. A request from a domain object is received for a retrieval of a portion of the data in the database in operation 15806. The data retrieval mechanism is utilized in operation 15808 to retrieve the portion of the data from the database. The portion of the data is passed through the data handler to the domain object in operation 15810." Column 275, Line 56); updating the view context object with the handler object i.e. "an application may be designed to retrieve the tax rate for the State of Illinois. When the user enters "Illinois" on the screen, the application first validates the user's entry by checking for its existence on the "State Tax Table", and then retrieves the tax rate for Illinois. Note that codes tables provide an additional degree of flexibility. If the tax rates changes, the data simply needs to be updated; no application logic needs to be modified." Column 102, Line 33); and refreshing the view object from the view context object (i.e. "A retrieveData( ) message might also be required, if the object model pre-instantiates objects before retrieving them. Similarly, refresh( ) could be used: the business object, if dirty, replaces any changes with data originally from the data store." Column 296, Line 13).*

### ***Conclusion***

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US006076092A                    Goldberg et al.

Teaches a system and process for providing improved database interfacing using query objects.

US006429882B1                    Abdelnur et al.

Teaches a user interface component in a business environment.

US006662188B1                  Rasmussen et al.

Teaches a metadata model in a business environment.

US005960410A                  Halpern

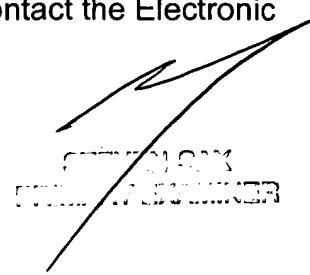
Teaches a device and method for object-based development of business applications software.

***Inquiry***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Boris Pesin whose telephone number is (703) 305-8774. The examiner can normally be reached on Monday-Friday except every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine Kincaid can be reached on (703) 308-0640. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Boris Pesin  
Kristine Kincaid